**CSCI 55700- Image Processing & Computer Vision**

**Semester Project**

**Face Recognition Using Eigenfaces**

**Abstract:**

The Face Recognition begins as the recognizing the query face from the set of known faces that are stored in the database. The Face recognition is done using the Eigenfaces algorithm that is used to extract the large amount of useful information from the reduced dataset. The Eigenfaces is used to effectively extract the discriminative information from the faces and match it with the face database which can reduce the complexity of the computation and time. The face recognition system provides an approach to identify the human faces, and it recognizes the person by comparing the characteristics of face to known faces that are trained and stored in the face database. The face images are projected on the feature space (face space), and this feature space is defined by the Eigen faces which are the Eigen Vectors of the set of the faces that are trained and stored in the database. The face recognition system will provide the ability to learn to recognizes the faces. This is a two-dimensional recognition system that takes the advantage of the 2-D characteristics views.

**Introduction:**

Face recognition is a quite complex task, because the faces are multidimensional, meaningful visual stimuli even small change in the computation will results in the loss of the meaningful information for recognizing the face. The Eigenfaces algorithm is used to reduce the computation in recognizing the face by eliminating the unnecessary information, such as the zero vectors and the algorithm will use the reduced small discriminative information that is crucial in the recognizing the information by reducing the computation complexity. The set of the known face images are taken in the database and then each face pixels are taken in the columned matrix and the single matrix will contain the all the pixels of the database of faces and then the computations are performed and the Eigen faces are calculated. The Eigen faces are the principal component and to find the eigen faces we use the principal component analysis method. If we multiple the training set with the eigen faces then we get the original image, the eigen faces are the characteristic feature of a face. The weights of the query face will give the degree of the eigen faces present in the image.

The input query image is matched with the database of faces and if the query image is from one of the faces from the database then the algorithm will return the exact match of the face.

First the database of faces is trained and the weights are computed and then the input query face is introduced and the weights of the query face is calculated and it is compared with each face in the database. If the difference in the weights is equals to zero then the system will treat it as the match in the face and it return the matched face. The approach transforms the face image onto the small set of characteristic feature images, called the “Eigen Faces”, which are initial principal component in training the database of face image. Recognition is performed by projecting the new query face image on the feature space and then the positions of the query face is compared to the position of the database of faces in order to recognize the face.

Eigen face are used because it is relatively simple, fast, accurate when compared to the other recognition algorithms in constrained environments. Which is insensitive to the small or gradual change in the face image.

**Literature Survey:**

For the Face recognition, I have taken the idea of the Eigenface Algorithm from various sources like research papers, tutorials and web pages. The project was mainly referred from the paper published in 1991 by the Mathew A. Turk and Alex P. Pentland.

For the technical implementation part of the project, I have mainly followed the above-mentioned research paper and the tutorials videos and the various slides.

<http://www.vision.jhu.edu/teaching/vision08/Handouts/case_study_pca1.pdf>

<http://www.face-rec.org/algorithms/pca/jcn.pdf>

<http://www.ijetae.com/files/Volume3Issue5/IJETAE_0513_14.pdf>

<https://www.cs.ucsb.edu/~mturk/Papers/jcn.pdf>

<http://cmp.felk.cvut.cz/cmp/courses/recognition/Labs/pca/kimo.pdf>

<https://www.youtube.com/watch?v=lnS9oCMO9NM>

<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

<http://www.cs.unc.edu/~lazebnik/spring09/lec22_eigenfaces.pdf>

<http://www.scholarpedia.org/article/Eigenfaces>

<http://openbio.sourceforge.net/resources/eigenfaces/eigenfaces-html/facesOptions.html>

<https://www.youtube.com/watch?v=SaEmG4wcFfg>

<https://www.youtube.com/watch?v=Kydde6XnFkk>

<https://www.youtube.com/watch?v=_lY74pXWlS8>

<https://www.youtube.com/watch?v=LYgBqJorF44>

<https://tinyurl.com/mhjbj43>

**Technical Description:**

The Eigen Faces algorithm (Principal Component Analysis) is used to extract the large amount of information from the reduced dataset.

1. I have taken the images from database of image faces. As part of this, I have taken the each of the image from a database and stored all the pixel value in a single columned matrix or vector. [columned matrix= 10,304x1]. Similarly, I have taken all the images from database and stored in a columned matrix.
2. Then, I have concatenated the all the columned matrix of the image into a single matrix ‘A’ [ 10,304x400] where 10,304- number of pixels in an image, 400- total number of images (training faces).
3. Then I have calculated the Eigen faces. Firstly, I have taken the sum of all the columns of ‘A’, and take the average face. (Avg=sum/400) of order 10,304x1 (µ= 1/M ( the input query image is taken and I have calculated the set of weights.
4. In order to reduce the higher dimensionality space to lower dimensionality space, I have calculated the Adjust matrix. (A- Avg) that by subtracting the average face from each face image matrix. Each column of resultant matrix is the adjusted face image.
5. Then, I have calculated the Covariance matrix. C M- number of images (400). Øi – 10304x400, so (Øi) x(Øi)’ result in 10304x10304 which is huge contains many unwanted zero vectors. Instead, I have taken (Øi)’ x(Øi) which gives 400x400 reduced matrix which is an Eigen vectors of the original input faces or Eigen faces.
6. In order, to compute the Eigen vectors from the covariance matrix, I have used the built-in function. ([V, D] =eig()) where V- Eigen vectors, D- Eigen Values.
7. I have calculated the Eigen faces by multiplying the Eigen vector with covariance matrix. (V x Ø). Each column of the resultant matrix is an Eigen face or Eigen Vector of the database of face images.
8. I have projected the input query images with correspond to each of the eigen faces. I have calculated the weights using Wi= (Ui)’ (I -µ). Where Ui -Eigen faces, I- input image (query image), µ- average face. The set of weights gives the similarity of query face image with the database of faces. (trained faces)
9. Next, I have checked whether input query image is actually a face or not. For this, I have taken the distance between the face space and the weights of input image. Øf= , Øadj= (I- µ) where Ui- Eigen Faces, µ- average face, Wi- weights. Then, the distance is squared to get d^2= (Øf - Øadj)

If the distance is less than the threshold then the input image is a face. The threshold is a maximum distance between the weights of the input image and the face space.

1. If the input is a face then all the database of faces is considered to be a input face image and set of weights are calculated for each face. If the Ei=(Wq-Wi) =0 or minimum value after the difference of weights is calculated is considered to be a match in the face.

**AT & T Face Database:**

I have used the AT& T face database that is the standard database for the face recognizing system, because the face in this database are taken under different illuminations and different pose in the faces. This database consists of the 40 people faces with the 10 different poses of each person totally 400 faces, and all images are of same size i.e., 112x92.

**Programming tool and Language**: MATLAB

**Experimental Results:**

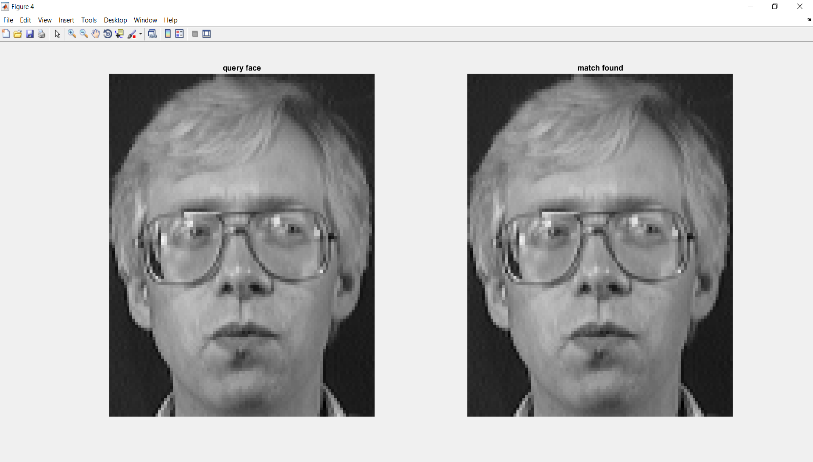
The implementation is getting the 100% of the success rate in recognizing the correct match in the face. Out of 20 attempt (sample data) of the trails of the recognizing the face 20 times the face was matched to the correct face.

Predicted = 100%, Actual = 100%

Trained Samples: 400, Tested sample: 40

Correct matches:









**Results:**

The Face recognition system is having small problem while recognizing the faces with the slight change in the pose of the face.

Strengths: The face recognition is fast in the computation and there are no zero vectors and that are eliminated that helps to reduce the time and space complexity in computation.

Weaknesses:

Due to the fastness of computation of the algorithm it has the chance of getting mismatch of input query face with another face even there is small change of the pose. If the weights of the input query face and the other two trained face matches then there is chance of getting the mismatch.

**Libraries and Helper Code:**

I have used the eig() function to find the eigen values and the eigen vectors from the covariance matrix.

I have used the helper code to read the databases of face images from the folder and sub directory.

**Conclusion:**

The Eigen faces algorithm is computationally fast and there are some drawbacks in recognizing, when there is slight change in the view of the face there is chance of mismatching due to the fast computation. The Eigenfaces method is sensitive to noise, illumination, expression, pose change in the face images.

**Future Work:**

In future, I will implement the face recognition system that take the faces in real-time and if the unknown face is given as the input query image then face match is not found from the known database of faces then the unknown query face image will be added to the database of faces and the process is repeated the until the unknow input query face is matched. I will extend the project to Autofocus because when the face image is captured by the camera there is trad off between the camera aperture and the focus length as a result the face in image will be blurred and it result in the loss of the discriminative information of the face and it can lead the face to get mismatch with the other face. So, to overcome this problem with the camera aperture I will use the Generalized Autofocus algorithm to deblur the face image and to find the depth of field for the capturing the sharpest point in the face image. If I get a chance to do this project again, I will use the Histogram of Oriented Gradients algorithm, because this algorithm will be efficient in extracting the features of the faces like directions, and magnitude of edge pixels of the face which results in the precise recognition of input query face with trained set of faces and it uses the machine learning algorithm(SVM) for predicting the local feature descriptors with in the classifier.

**References:**

[1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Maui, HI, 1991, pp. 586-591.  
doi: 10.1109/CVPR.1991.139758

[2] V. P. Kshirsagar, M. R. Baviskar and M. E. Gaikwad, "Face recognition using Eigenfaces," 2011 3rd International Conference on Computer Research and Development, Shanghai, 2011, pp. 302-306.  
doi: 10.1109/ICCRD.2011.5764137

[3] A. Pentland, B. Moghaddam and T. Starner, "View-based and modular eigenspaces for face recognition," 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA ,1994, pp. 84-91.

doi: 10.1109/CVPR.1994.323814

[4] M. H. Yang, N. Ahuja and D. Kriegman, "Face recognition using kernel eigenfaces," Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*,* Vancouver, BC, 2000, pp.37 -40 vol.1.  
doi: 10.1109/ICIP.2000.900886

[5] T. Ahonen, A. Hadid and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037-2041, Dec. 2006.  
doi: 10.1109/TPAMI.2006.244

[6] Jian Yang, D. Zhang, A. F. Frangi and Jing-yu Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 1, pp. 131-137, Jan. 2004.  
doi: 10.1109/TPAMI.2004.1261097